



REALTEXT AUTHORIZING GUIDE

RealSystem Release 8

15 December 2000

RealNetworks, Inc.
2601 Elliott Avenue, Suite 1000
Seattle, WA 98121
U.S.A.

<http://www.real.com>
<http://www.realnetworks.com>

©1998-2000 RealNetworks, Inc. All rights reserved.

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of RealNetworks, Inc.

Printed in the United States of America.

RealNetworks, the Real Bubble, RealSystem, RealSystem G2, RealServer, RealPlayer, RealJukebox, RealProducer, RealProxy, RealAudio, RealVideo, RealText, RealPix, RealPresenter, RealPublisher, SureStream, Real Accessory Zone, RBN, Real Broadcast Network, Real.com, Real.com Guide, Real.com Take 5, Real G2 with Flash, RealMP, RealSites, RealSlideshow, and RealStore.com are trademarks or registered trademarks of RealNetworks, Inc.

Other products and corporate names are the trademarks or registered trademarks of their respective companies.



CONTENTS

RELEASE UPDATES	1
Updates for RealSystem 8	1
Updates For RealSystem Release 7, Update 1	1
Support for Korean Text	1
Multiple Windows with RealPlayer 7	2
UNIX Font Mappings	2
Updates For RealSystem Release 7	2
New Features	2
Versioning Supported in Head Tag.....	2
Support for mac-roman Character Set.....	3
New Default Character Set of iso-8859-1 in Version 1.2 ..	3
Changes and Corrections.....	3
Clarification for the <clear/> Tag	3
RealText Broadcast Application Moved to Authoring Kit. 3	
Upper 128 Characters in the mac-roman Character Set... 3	
Revisions Introduced December 21, 1998.....	4
INTRODUCTION	5
Tools for Creating RealText.....	5
Conventions Used in this Manual.....	6
Additional RealSystem Resources.....	6
Technical Support.....	7
1 REALTEXT AUTHORIZING	8
Choosing Window Types	8
Using RealText Mark-Up.....	9
Creating a RealText Presentation.....	9
Notes on Text Streaming.....	11
2 WINDOW TAG ATTRIBUTES	13
Specifying Window Attributes	13
type="window type"	13
duration="dd:hh:mm:ss.xyz"	14
width="pixels"	14
height="pixels"	15

	bgcolor="color"	15
	scrollrate="pixels per second"	15
	crawlrate="pixels per second"	15
	link="color"	15
	underline_hyperlinks="true false"	16
	wordwrap="true false"	16
	loop="true false"	16
	extraspaces="use ignore"	16
	version="version number"	16
3	TEXT TAGS	18
	Time and Position Tags	18
	<time begin/> and <time end/> Tags.....	18
	Tips for Using <time/> Tags.....	19
	<clear/> Tag	19
	<pos x/> and <pos y/> Tags.....	20
	<tu> and <tl> Tags.....	21
	Layout Tags	21
	<p>...</p>	21
	 	21
	21
	22
	22
	<hr/>.....	22
	<center>...</center>.....	22
	<pre>...</pre>	22
	Appearance Tags.....	23
	23
	<i>...</i>	23
	<s>...</s>	23
	<u>...</u>	23
	 Tag.....	23
	bgcolor="color"	24
	charset="character set"	24
	color="color"	25
	face="font name"	25
	English and European Language Fonts	25
	Asian Language Fonts	27
	size="n"	28
	Working with the Font Tag	29
	Command Tags	29
	Creating a Mail Link.....	29

	Opening a URL in RealPlayer or a Browser	30
	Issuing RealPlayer Commands	31
	Seeking Into a Presentation.....	31
	Pausing a Presentation	31
	Resuming Playback	31
	Popping Up New RealPlayer Windows.....	31
	name.....	32
	URL.....	32
	playmode	32
	Examples of Opening New RealPlayer Windows.....	33
	Ensuring Text Delivery under any Circumstance	34
	Colors	34
	Transparency	35
	Coded Characters	35
	Using the mac-roman Character Set.....	36
4	EXAMPLES	37
	Generic Window	37
	TickerTape Window	38
	Scrolling News Window.....	39
	TelePrompter Window.....	40
A	SUMMARY OF WINDOW TAG ATTRIBUTES	42
B	SUMMARY OF TEXT TAGS	44



RELEASE UPDATES

If you are new to RealText, start with “Introduction” on page 5. If you have worked with RealText before this release, this section gives you a quick look at the changes included in this version of the manual.

Updates for RealSystem 8

RealSystem 8 Gold includes RealText version 1.5. This version supports hyperlinks in the format *protocol:path*. For more information, see “Opening a URL in RealPlayer or a Browser” on page 30.

Note

To ensure that RealPlayers can use these features, include `version="1.5"` in the `<window.../>` tag of each RealText file. This causes earlier RealPlayers to update to the latest RealPlayer through the autoupdate feature.

Updates For RealSystem Release 7, Update 1

The following sections describe new RealText features for RealSystem Release 7, Update 1, which includes RealText version 1.4. RealPlayer 7 or higher is required to play RealText clips that use the following new features. To stream clips using the following new features, you need RealServer 7 with Update 1 (or a higher version of RealServer).

Support for Korean Text

RealText 1.4 now supports the Korean character set iso-2022-kr. The following Korean fonts are available:

- AppleGothic
- Batang

- BatangChe
- Gothic
- Gulim
- GulimChe

Korean is currently supported on Windows and Macintosh, but not UNIX-based RealPlayers.

Additional Information

See “ Tag” on page 23.

Multiple Windows with RealPlayer 7

RealPlayer 7 or higher supports multiple windows. You can launch a presentation in a new, pop-up window through a RealText hyperlink. See “Popping Up New RealPlayer Windows” on page 31 for more information.

UNIX Font Mappings

The table “RealText Font Support for us-ascii and iso-8859-1 Character Sets” on page 26 lists the default fonts used for RealPlayer on UNIX systems. These fonts are used if the system does not have the specified Windows fonts available.

Updates For RealSystem Release 7

The following sections describe new features in RealText for RealSystem Release 7. They also list changes and corrections to this authoring guide. Release 7 includes RealText version 1.2.

New Features

RealPlayer 7 or higher is required to play RealText clips that use the following new features. (RealPlayer users can get the latest RealPlayer through the autoupdate feature.) Unless otherwise noted, the clips can be streamed from RealServer G2 Gold with update 2, or any higher version of RealServer.

Versioning Supported in Head Tag

The <window> tag can now include a version=“version number” attribute that causes RealPlayer to use autoupdate to upgrade its RealText plug-in. A version

number is also needed for character sets that include accented characters. See page 16 for more information.

Support for mac-roman Character Set

RealText now supports the mac-roman character set for entering accented European languages on a Macintosh. See “ Tag” starting on page 23 for more information.

New Default Character Set of iso-8859-1 in Version 1.2

If you specify RealText version 1.2 or higher in the <window> tag, the default character set iso-8859-1 is used instead of us-ascii. This character set is the same as us-ascii, but includes support for accented characters commonly used in European languages. See “ Tag” starting on page 23 for more information.

Changes and Corrections

Clarification for the <clear/> Tag

Information on the <clear/> tag has been clarified to indicate that the tag erases text that has an end time that has not elapsed. See page 19.

RealText Broadcast Application Moved to Authoring Kit

The RealText broadcast application has been removed from the HTML version of this authoring guide. The broadcast application is now part of the RealSystem Authoring Kit, available through registration at **<http://www.reálnetworks.com/products/authkit/index.html>**.

Upper 128 Characters in the mac-roman Character Set

In the mac-roman character set, the upper 128 coded characters differ from those in the default iso-8859-1 character set. See “Using the mac-roman Character Set” on page 36 for information on using the upper 128 characters with mac-roman.

Revisions Introduced December 21, 1998

Recommendations for Scrollrate and Crawlrate

See the information on the <window> tag scrollrate and crawlrate attributes on page 15 for tips on setting these rates.

Time Tag Tips

A few tips on using <time.../> tags have been added.

Updated Font Information

Information about the tag on page 23 has been updated. The end of that section contains examples of the tag in use.



INTRODUCTION

Welcome to RealText, a RealSystem product for streaming text from files or live sources. With RealText you can create presentations consisting of text alone, or combine text with other media to create, for example, closed-captioned video. This guide tells you how to use the RealText mark-up language to format streaming text for playback in RealPlayer.

Tip

Download the HTML version of this guide from **<http://service.real.com/help/library/encoders.html>** to get RealText examples you can view with RealPlayer.

Tools for Creating RealText

You need the following tools to create and test your RealText presentation:

- Text Editor

To create a RealText file, you can use any word processor, text editor, or XML editor that can save output as plain text.

- RealPlayer

Use RealPlayer, available free at **<http://www.real.com>**, to test your RealText presentation. Other applications may also have RealPlayer features that enable them to receive RealText as well. Versions of RealPlayer earlier than RealPlayer G2, such as RealPlayer 4.0 and 5.0, cannot display RealText.

- RealServer or a Web server

RealServer streams RealText presentations to RealPlayer. If you are not operating RealServer yourself, you need to have access to RealServer through, for example, an Internet Service Provider (ISP). Note that previous versions of RealServer cannot stream RealText. A standard Web

server can also download a static RealText clip to RealPlayer. It cannot broadcast RealText live, however.

- **Broadcast Application**

A broadcast application can capture live text, add RealText mark-up to it, and send it to RealServer. A sample broadcast application is included with the RealSystem Authoring Kit, which is available for download at <http://www.realnetworks.com/products/authkit/index.html>. You can also build a broadcast application with the RealSystem Software Development Kit (SDK).

Conventions Used in this Manual

The following table explains the conventions used in this manual.

Notational Conventions	
Convention	Meaning
<i>variables</i>	Italicized text represents variables. Substitute values appropriate for your situation.
[options]	Square brackets indicate optional values you may or may not need to use.
choice 1 choice 2	Vertical lines separate values you can choose between.
...	Ellipses indicate nonessential information omitted from the example.

Additional RealSystem Resources

In addition to this manual, you may need the following RealNetworks resources, available at <http://service.real.com/help/library/encoders.html>:

- *RealSystem Production Guide*

This manual explains the basics of streaming files with RealSystem. It tells how to calculate bandwidth needs and shows how to put a multimedia presentation together.

- *RealText Broadcasting*

This technical blueprint explains how to use a simple application to broadcast RealText. The broadcast application and technical blueprint are

included in the RealSystem Authoring Kit, available at
<http://www.realnetworks.com/products/authkit/index.html>.

- *Embedded RealPlayer Extended Functionality Guide*

This guide supplements *RealSystem Production Guide*. It explains how to use JavaScript or VBScript to control RealPlayer functions for a presentation embedded in a Web page.

- *RealServer Administration Guide*

The basic reference for the RealServer administrator, this manual explains how to set up, configure, and run RealServer to stream multimedia. You need this manual only if you are running RealServer yourself. Download it from **<http://service.real.com/help/library/servers.html>**.

- RealSystem Software Development Kit (SDK)

The RealSystem SDK lets you integrate applications with RealSystem. You need the SDK to build a customized RealText broadcast application, for example. A knowledge of programming is required to use the SDK. Download the SDK from **<http://www.realnetworks.com/devzone/>**.

Technical Support

For technical support with RealText, please fill out the form at:

- **http://customerrelations.real.com/scripts/rnforms/contact_tech_service.asp**

The information you provide in this form will help technical support personnel to give you a prompt response. For general information about RealNetworks' technical support, visit:

- **<http://service.real.com/help/call.html>**

Chapter 1

REALTEXT AUTHORIZING

With RealText, you can create text presentations that you stream alone or in combination with other media such as audio or video. You simply create a RealText file with mark-up similar to HTML to describe how and when the text displays. RealServer or a Web server can then stream the text to RealPlayer or Web browsers.

Choosing Window Types

RealText provides a number of window styles that you can choose depending on how you want to display text:

- generic

A generic window has no preset parameters. You can use it to create any RealText display allowed by the RealText mark-up. You can display and erase lines of text, scroll text through the window, or have text crawl from side to side, for example.

- ScrollingNews

A ScrollingNews window is preset to have text scroll from the bottom of the window to the top at a set rate for the entire presentation. The text does not crawl from side to side, though.

- TickerTape

Text in a TickerTape window crawls from the right side of the window to the left. It can also loop back around to the right. It does not scroll up or down, however. Text displays next to the window's top or bottom edge.

- Marquee

The Marquee window is like the TickerTape in that text crawls from right to left and can loop. It is different in that text is centered vertically within the window.

- TelePrompter

A TelePrompter window behaves like a generic window except that text arriving at the bottom edge of the window causes the text above it to move up just enough to display the new line.

Using RealText Mark-Up

The RealText mark-up language is similar to HTML. If you are familiar with HTML, you will pick up RealText quickly. The mark-up has some important differences from HTML, though. Keep the following points in mind when writing a RealText file:

- Use lowercase characters for RealText tags and attributes. This keeps your RealText file compliant with the SMIL mark-up language and its XML parent language.
- A tag that does not have a corresponding end tag (for example, the `` tag has the end tag ``), closes with a forward slash. For example:
`
`
- Attribute values must be enclosed in double quotation marks.
- Save your RealText file with the file extension `.rt`. Do not include spaces in the file name. For example, you can have the file `my_realtext.rt` but not the file `my realtext.rt`.
- Use codes to include angle brackets, ampersands, or nonbreaking spaces as RealText display characters. See “Coded Characters” on page 35.
- As in HTML, you can add a comment to a RealText file like the following. Note that the comment tag does not need to close with a slash.
`<!-- This is a comment -->`

Additional Information

To learn more about XML, the parent language for RealText, visit <http://www.w3.org/XML>.

Creating a RealText Presentation

The following steps describe how to create a static RealText file. The rest of this manual describes the RealText mark-up in detail. The technical blueprint *RealText Broadcasting* explains how to use a simple application to broadcast

RealText. The broadcast application and technical blueprint are included in the RealSystem Authoring Kit, available as a download at <http://www.realnworks.com/products/authkit/index.html>.

► To Create a RealText File:

1. Open a new file in a text editor or XML editor. At the top of the file, add the <window> tag with necessary options. Add the </window> tag at the bottom:

```
<window...options...>  
...All Other Mark-up Goes Between These Tags...  
</window>
```

Additional Information

See “Window Tag Attributes” on page 13.

2. Between the <window> and </window> tags, add the text that will display in RealPlayer. Format the text with the text tags:

```
<window>  
Mary had a little lamb,  
<br/><time begin="3"/>little lamb,  
<br/><time begin="6"/>little lamb,  
<br/><time begin="9"/>Mary had a little lamb  
<br/><time begin="12"/>whose fleece was white as snow.  
</window>
```

Additional Information

See “Text Tags” on page 18.

3. Save the file as plain text, using the .rt extension to mark the file as a RealText document. On your local machine, open the RealText file with RealPlayer to test the presentation.

Tip

Playing a RealText file on your local machine indicates whether the RealText mark-up is correct. However, it does not guarantee that the file will stream across a network well. Be sure to test that the presentation streams correctly from RealServer.

4. To combine RealText with another file, create a SMIL file that controls the overall presentation. For example, the SMIL file can list a RealText file and video file played together:

```
<smil>
  <body>
    <par>
      <textstream src="rtsp://realserver.company.com/mary.rt"/>
      <video src="rtsp://realserver.company.com/mary.rm"/>
    </par>
  </body>
</smil>
```

Additional Information

See *RealSystem Production Guide* available at

<http://service.real.com/help/library/encoders.html>

for information about creating a SMIL file.

5. Move your files to RealServer. If you are using an Internet Service Provider, for example, contact the ISP's RealServer administrator for instructions on doing this.
6. In your Web page, add a hypertext link to the SMIL file. Or, if the RealText file is the only file in your presentation, simply link to that file.

Additional Information

RealSystem Production Guide explains the options for linking your Web page to your presentation. It also explains how to play your RealText presentation in your Web page rather than in RealPlayer.

7. Test the presentation by clicking the hyperlink in your Web page. This launches RealPlayer, which displays the streaming text with the properties you defined through the RealText mark-up.

Notes on Text Streaming

1. RealText consumes minimal bandwidth, typically less than 1 Kbps. RealText presentations are therefore easily accessible to users with slow network connections. When combining RealText with another media type, structure the presentation so that RealText has approximately 1 Kbps of available bandwidth.

Additional Information

See *RealSystem Production Guide* available at

<http://service.real.com/help/library/encoders.html>

for more information about bandwidth consumed by video or audio that accompanies RealText.

2. Although RealServer provides reliable streaming, packets occasionally may be lost. If a block of text does not get through, RealPlayer displays the following to indicate missing text:

...

Chapter 2

WINDOW TAG ATTRIBUTES

The `<window>` and `</window>` tags that begin and end a RealText file, respectively, determine the window type and set attributes such as window height and width. This chapter explains the options you can set.

Note

When you are familiar with the attributes, you can refer to “Summary of Window Tag Attributes” on page 42.

Specifying Window Attributes

You specify attributes within the `<window>` tag, much as you specify HTML table attributes within the HTML `<TABLE>` tag. For example:

```
<window type="tickertape" duration="2:05:00.0" underline_hyperlinks="false">
```

No attributes are required for the `<window>` tag, however. If you do not specify an attribute, the attribute's default value applies. When you include an attribute, enclose its value in double quotation marks.

type=“window type”

The `type=“window type”` attribute defines the RealText window type:

- generic
- tickertape
- marquee
- scrollingnews
- teleprompter

The default is `type=“generic”`.

Additional Information

For descriptions of the window types, see “Choosing Window Types” on page 8.

duration=“dd:hh:mm:ss.xyz”

The duration attribute specifies the time, relative to the start of the presentation, that this RealText stream stops playing. The default is 60 seconds. The RealPlayer timing slider is keyed to this value, which is in 24-hour format, where dd is days, hh is hours, mm is minutes, ss is seconds, x is tenths of seconds, y is hundredths of seconds, and z is milliseconds.

Only the ss value is required. When the time value does not include a decimal point, the last field is read as the seconds. For example, 2:05 means 2 minutes and 5 seconds, whereas 2:05:00 means 2 hours and 5 minutes. You can also specify just the seconds. For example, both of the following values end the text stream 2 hours and 5 minutes after the stream begins:

```
duration="2:05:00.0"
```

```
duration="7500"
```

Tip

Set a high duration when you start building a RealText presentation. Set the final duration time when you have finished defining the mark-up.

width=“pixels”

The width attribute determines the window width in pixels. The default is 500 for TickerTape and Marquee windows, 320 for other window types. SMIL layout tags can specify a playback region width that overrides the width set here. If word wrap is on, the line length for wrapping corresponds to this width value, not the actual region width. Text centering, however, corresponds to the actual region width set in the SMIL file.

Additional Information

See *RealSystem Production Guide* available at

<http://service.real.com/help/library/encoders.html>
for information about SMIL.

height=“pixels”

The height attribute sets the window height in pixels. The default is 30 for TickerTape and Marquee windows, 180 for other window types. SMIL layout tags can specify a window height that overrides the height set here. This typically does not adversely affect the presentation, however.

bgcolor=“color”

The bgcolor=“color” attribute determines the window's background color. The default is black for TickerTape windows and white for all other window types. See “Colors” on page 34 for information on valid colors.

scrollrate=“pixels per second”

The scrollrate attribute sets the number of pixels per second that the text moves vertically. It has no effect on TickerTape and Marquee windows. The default is 10 for ScrollingNews windows and 0 for all other window types.

Tip

For best results, use a scrollrate under 30. (Best values are 25, 20, 10, 8, 5, 4, 2, and 1.) For rates faster than 30, use multiples of 20 or 25, such as 40, 50, 60, 75, 80, and so on.

crawlrate=“pixels per second”

The crawlrate attribute specifies the number of pixels per second that the text moves horizontally. The default is 20 for TickerTape and Marquee windows, 0 for other window types.

Tip

For best results, use a crawlrate under 30. (Best values are 25, 20, 10, 8, 5, 4, 2, and 1.) For rates faster than 30, use multiples of 20 or 25, such as 40, 50, 60, 75, 80, and so on.

link=“color”

The link=“color” attribute sets the color of hyperlinks within the text. The default is blue. See “Colors” on page 34 for other color options.

underline_hyperlinks="true|false"

The `underline_hyperlinks="true|false"` attribute determines whether hyperlinks are underlined. The default is true.

wordwrap="true|false"

The `wordwrap="true|false"` attribute, which defaults to true, specifies whether word wrap is performed. When word wrap is on, text lines longer than the specified window width wrap to the following line. If it is off, long lines are truncated by the window border. This attribute has no effect for windows that have horizontal text motion, such as the TickerTape.

loop="true|false"

The `loop="true|false"` attribute is available only in TickerTape and Marquee windows, where it defaults to true. When set to true, this attribute tells RealPlayer to buffer all text and redisplay ("loop") it if and when the stream runs dry, which occurs when the text has moved out of the window and no new text has arrived. If the text has looped and new text arrives, the new text displays as soon as the old text has moved out of the window. The new text then becomes part of the loop.

extraspaces="use|ignore"

The default value "use" for the `extraspaces="use|ignore"` attribute makes RealText recognize all blank spaces between mark-up tags. If three spaces occur between two words in the RealText file, for example, RealPlayer displays all three spaces. It also recognizes carriage returns and tabs.

If you specify `extraspaces="ignore"`, RealPlayer treats spaces, tabs, line feeds, and carriage returns as does a Web browser, except when they are between the `<pre>...</pre>` tags described on page 22. When spaces or carriage returns occur contiguously in the text, RealPlayer interprets them as a single space, no matter how many of them are present. So in this case the three spaces display as one space in RealPlayer. It treats each tab as a single space, though.

version="version number"

The `<window>` tag can include a version number:

```
<window version="1.5" ...>
```

You typically do not have to specify a version number when using RealText in English. Properly displaying languages other than English may require that you specify a version number explicitly in the <window> tag, however. This manual tells you when a version number is required.

Because newer versions of RealText encompass all features from previous releases, you can specify a higher RealText version number. If a feature requires RealText version 1.2, for example, but the latest RealText version is 1.5, you can use 1.5 as the version number.

Note

Including the current RealText version number causes RealPlayer to autoupdate to the latest version of the RealText plug-in if necessary.

Chapter 3

TEXT TAGS

RealText provides many mark-up tags that define how the streaming text looks and operates. A tag's default value applies if you do not specify a tag value. You can place mark-up tags anywhere on a line.

Note

When you are familiar with the tags, you can refer to “Summary of Text Tags” on page 44.

Time and Position Tags

The following tags affect when and where the text appears within the window.

<time begin/> and <time end/> Tags

The <time/> tags control the RealText presentation timeline by determining when a text component appears and disappears, respectively, relative to the start of the presentation:

```
<time begin="dd:hh:mm:ss.xyz"/>
```

```
<time end="dd:hh:mm:ss.xyz"/>
```

They are meant primarily for window text that does not scroll or crawl. If you do not specify begin times, RealPlayer displays all text as quickly as it can.

The <time/> tag values are in 24-hour format, where dd is days, hh is hours, mm is minutes, ss is seconds, x is tenths of seconds, y is hundredths of seconds, and z is milliseconds. Only the ss field is required. When the time value does not include a decimal point, the last field is read as the seconds. For example, 1:30 means 1 minute and 30 seconds, whereas 1:30:00 means 1 hour and 30 minutes. Note that all the following values are equivalent. They all start the text component 90 minutes after the stream begins:

```
<time begin="1:30:00.0"/>
```

```
<time begin="90:00"/>
```

```
<time begin="5400"/>
```

Text with an end value is erased when the specified end value is reached. Otherwise it stays active until the presentation ends or the entire window is erased with `<clear/>`. Note that you can combine the begin and end attributes in a single `<time/>` tag as shown here:

```
<time begin="23" end="55.5"/>This text displays 23 seconds into the presentation
and disappears at 55.5 seconds.
```

All text following a `<time/>` tag has the specified begin and end values until new values are given. Once you specify an end time for a text component, you must specify an end time for all following components. For example, the following text would not display properly:

```
<time begin="23" end="55.5"/>Display at 23 seconds in.
<time begin="56"/>Display at 56 seconds in.
```

Because the second line does not include an end value, the previous end value of 55.5 still applies. The second line cannot be displayed because its begin time is later than its end time.

Tips for Using `<time/>` Tags

Here are some times on using `<time/>` tags:

- The `<time/>` tags are not necessary in a window with a non-zero scroll rate or crawl rate unless you want text to become visible after it has moved into the window, or to disappear before it moves out of the window.
- To freeze text on the screen after the `<window>` tag's duration has elapsed, do not set an end time. Or, have the end time exceed the window's duration:

```
<window duration="30">
...(some text elements)...
  <time begin="25" end="31"/>Text that stays frozen onscreen.
</window>
```

- To replace a line of text with a new line every few seconds (as in video subtitles), do not use end times. For each new line of text, set the appropriate begin time followed by a `<clear/>` tag.

`<clear/>` Tag

This tag clears the existing text buffers to remove all text from the window:

```
<clear/>
```


The text that follows this tag is then displayed starting at the window's normal starting point. In a window that does not scroll or crawl, you can add `<clear/>` after `<time begin="..."/>` to erase existing text when new text arrives. For example, you would specify the following to clear old text and display "Hello!" at 3 minutes into the stream:

```
<time begin="3:00"/><clear/>Hello!
```

A `<clear/>` tag removes all preceding text, even text that has an end time that has not yet elapsed. Consider this example:

```
<time begin="5"/>They all lived happily.  
<time begin="10" time end="20"/>And so our story ends.  
<time begin="15"/><clear/>Goodbye!
```

The second line of text is set to end at 20 seconds. The `<clear/>` tag appears 15 seconds into the presentation and clears this line, however.

Note

The `<clear/>` tag does not reset text appearance. For example, if text appears bolded before the `<clear/>` tag, it remains bolded after the `<clear/>` tag.

<pos x/> and <pos y/> Tags

These tags position the text horizontally and vertically, respectively:

```
<pos x="pixels"/>
```

```
<pos y="pixels"/>
```

The `<pos y/>` tag moves the upper, left corner of the subsequent text block the specified number of pixels down from the window's top edge. The `<pos x/>` tag indents the text block the specified number of pixels in addition to the two-pixel default padding that applies to all text blocks. You can combine both tags in a single tag like this:

```
<pos x="10" y="55"/>
```

Note

These tags work only if scroll rate and crawl rate are both 0 (zero).

<tu> and <tl>Tags

These tags function only with TickerTape windows. They display the enclosed text at the window's upper (<tu>) or lower (<tl>) edge:

```
<tu [color="color"]>...</tu>
```

```
<tl [color="color"]>...</tl>
```

When a tag specifies a color with the color option, the color applies to text enclosed by all subsequent tags of that type until another tag of that type changes the color. However, color specified for <tu> elements does not affect color for <tl> elements, and vice versa.

Additional Information

Refer to “Colors” on page 34 for a list of available colors.

Layout Tags

Much as in HTML, the following tags let you define the layout of RealText in the RealPlayer window.

<p>...</p>

The <p>...</p> tags add space between text. In TickerTape and Marquee windows, they move the “cursor” to the right edge of the window. In all other window types, the <p> and </p> each cause the next text to display two lines down.

**
**

The
 tag adds space between text. In TickerTape and Marquee windows, it moves the “cursor” to the right edge of the window. In all other window types, this tag causes the text that follows to display on the next line.

...

The ... tags are for compatibility with HTML lists. Text between these tags is indented, but not numbered.

...

The ... tags are for compatibility with HTML lists. Text between these tags is indented, but not bulleted.

...

The ... tags are for compatibility with HTML lists. They act like a
 tag.

<hr/>

The <hr/> tag is for compatibility with HTML horizontal rules. It acts as two
 tags.

<center>...</center>

The <center>...</center> tags center the enclosed text. Text is centered according to the actual window width, which may differ from the width attribute. These tags behave the same as HTML centering tags, but they have no effect in windows with horizontal motion, such as TickerTape and Marquee windows. The <center> tag forces a line break if and only if a line break caused by a tag such as
, <p>, or <hr/> does not immediately precede it. The </center> tag always causes a line break.

Note

RealText does not center text until it has determined the line length. In rare instances, one streamed packet may contain the first part of the line while another packet received several seconds later contains the end of the line. In this case, the first part displays flush left, and the entire line is centered and redisplayed when the second packet arrives.

<pre>...</pre>

The <pre>...</pre> tags work the same as in HTML. Text tagged with <pre> uses the Courier font at the current size. For example, a preceding makes the preformatted text one size larger than the default font size. Line breaks, spaces, and tabs are preserved, with tabs defaulting to 64

pixels for 16 point text (the normal point size). Tab spaces are determined by dividing the text height by 2, then multiplying by 8.

Additional Information

For information on text heights, see the “Font Sizes” table on page 23. See also the `<window>` tag attribute `extraspaces="use|ignore"` on page 16.

Appearance Tags

You can use the following set of tags to change the appearance of text.

`...`

The `...` tags display the enclosed text **bolded**.

`<i>...</i>`

The `<i>...</i>` tags display the enclosed text *italicized*.

`<s>...</s>`

The `<s>...</s>` tags ~~strike through~~ the enclosed text.

`<u>...</u>`

The `<u>...</u>` tags display the enclosed text underlined.

`` Tag

The `` tag lets you specify text characteristics:

`...`

Like the HTML `` tag, the RealText `` tag uses an end tag, ``, and can contain several attributes as described in the following sections. For example:

`...text...`

bgcolor="color"

Use this tag attribute to set the text background color. The default is bgcolor="transparent".

Additional Information

See "Colors" on page 34.

charset="character set"

With this tag attribute, you can control the character set used to display the text. You can change character sets within a RealText file with a tag. The following are the character sets you can use.

- us-ascii
Default character set used with most RealText fonts when no RealText version is specified in the <window> tag.
- iso-8859-1
Default character set used when you specify version="1.2" or higher in the <window> tag. This character set is identical to us-ascii, but includes support for accented characters (upper 128 characters) used in European languages.
- mac-roman
Character set to use when entering an accented language on a Macintosh. It properly displays accent marks such as umlauts (for example, "ü") when they display on other platforms. You do not need to use this character set when displaying an unaccented language such as English, or writing RealText files for accented languages on a Windows or UNIX machine. Use version="1.2" (or higher if using a newer version of RealText) in the <window.../> tag to make all RealPlayers handle this character set correctly.
- x-sjis
Character set for Kanji and the Osaka font. Use version="1.2" (or higher if using a newer version of RealText) in the <window.../> tag to make RealPlayer handle Japanese text correctly.
- gb2312
Character set for Simplified Chinese.
- big5
Character set for Traditional Chinese.

- iso-2022-kr

Character set for Korean fonts. Use version="1.4" (or higher if using a newer version of RealText) in the <window.../> tag to make RealPlayer handle Korean text correctly.

Note

Not all computers are capable of displaying all character sets. If a computer does not recognize the specified character set, it displays the text in its default character set. The result is typically unreadable.

Warning

RealText always uses its default character set regardless of the default character set of the machine receiving the stream. You must explicitly specify a different character set to stream RealText to machines that do not render in RealText's default character set.

color="color"

This tag attribute lets you control the font color. It supports all color values available in HTML. For TickerTape windows, however, this attribute has no effect. The <tu> and <tl> tags set the TickerTape text colors.

Additional Information

See "Colors" on page 34.

face="font name"

This tag attribute controls the text font. Fonts correspond to character sets as described above. For non-Western fonts, you must specify the correct character set for the font to display properly.

English and European Language Fonts

When writing in English or European languages, use a font name from the "Windows Font Name" column of the following table, which lists fonts that use the us-ascii or iso-8859-1 character set. If you specify no font, RealText uses Times New Roman or Times regardless of the character set specified.

On Macintosh and UNIX, RealText uses the specified font if that font is available. If the font isn't available, RealText defaults to a system font as

indicated in the table below. If you use the Algerian font, for example, a RealPlayer on a Macintosh displays the text in that font as long as the font is available. Otherwise, it displays the text in Courier. In a few cases, RealText always defaults to a system font. These cases are indicated with the notation “(always)”. For example, the Fixedsys font always displays as Courier on a Macintosh.

RealText Font Support for us-ascii and iso-8859-1 Character Sets

Windows Font Name	Macintosh Default if Font not Available	UNIX Default if Font not Available
Algerian	Courier	Courier
Arial	Helvetica	Helvetica
Arial Black	Helvetica	Helvetica
Arial Narrow	Helvetica	Helvetica
Arial Rounded Mt Bold	Helvetica	Helvetica
Book Antiqua	Helvetica	Helvetica
Bookman Old Style	Helvetica	Helvetica
Braggadocio	Helvetica	Helvetica
Britannic Bold	Helvetica	Helvetica
Brush Script	Times	Times
Century Gothic	Helvetica	Helvetica
Century Schoolbook	Helvetica	Helvetica
Colonna Mt	Colonna Mt or Times	Times
Comic Sans Ms	Times	Times
Courier New	Courier	Courier
Desdemona	Helvetica	Helvetica
Fixedsys	Courier (always)	Courier
Footlight Mt Light	Helvetica	Helvetica
Garamond	Times	Times
Haettenschweiler	Helvetica	Helvetica
Helvetica (Arial is used if Helvetica is not found.)	Helvetica	Helvetica
Impact	Helvetica	Helvetica
Kino Mt	Times	Times
Matura Mt Script Capitals	Times	Times

(Table Page 1 of 2)

RealText Font Support for us-ascii and iso-8859-1 Character Sets

Windows Font Name	Macintosh Default if Font not Available	UNIX Default if Font not Available
Modern	Helvetica	Helvetica
Ms Dialog	Times	Times
Ms Dialog Light	Times	Times
Ms Linedraw	Helvetica	Helvetica
Ms Sans Serif	Helvetica	Helvetica
Ms Serif	Helvetica	Helvetica
Ms Systemex	Times	Times
Playbill	Times	Times
Small Fonts	Times	Times
System	Geneva (always)	Times
Terminal	Geneva	Times
Times New Roman	Times (always)	Times
Verdana	Helvetica	Helvetica
Wide Latin	Helvetica	Helvetica

(Table Page 2 of 2)

Tip

A Macintosh that has Microsoft's Internet Explorer 4.0 or later browser installed should have most of the Windows fonts available.

Asian Language Fonts

RealText also supports the following fonts that use character sets other than us-ascii and iso-8859-1.

RealText Font Support for Non-Western Character Sets

Font Name	Characters	RealText Font Face Tag	charset
AppleGothic	Korean		iso-2022-kr
Batang	Korean		iso-2022-kr
BatangChe	Korean		iso-2022-kr
Gothic	Korean		iso-2022-kr
Gulim	Korean		iso-2022-kr
GulimChe	Korean		iso-2022-kr

(Table Page 1 of 2)

RealText Font Support for Non-Western Character Sets (continued)

Font Name	Characters	RealText Font Face Tag	charset
Osaka	Kanji	<code></code>	x-sjis
Seoul	Korean	<code></code>	iso-2022-kr
， 宋体	Simplified Chinese	<code></code> (The face name displays as gibberish without the gb2312 character set.)	gb2312
細明體	Traditional Chinese	<code></code> (The face name displays as gibberish without the big5 character set.)	big5

(Table Page 2 of 2)

Note

Korean and Japanese are supported in RealPlayer for Windows and Macintosh, but not for UNIX.

size="n"

This `` tag attribute lets you control the font size. You can use relative sizes or absolute sizes as shown in the table below. This table also lists the height in pixels for each size. The pixel sizes are for reference only. You cannot specify a pixel size directly in RealText.

Font Sizes

Relative Size	Absolute Size	Pixel Size Reference
-2	1	12 pixels
-1	2	14 pixels
+0 (default)	3	16 pixels
+1	4	20 pixels
+2	5	24 pixels
+3	6	36 pixels
+4	7	48 pixels

Note

You can also specify relative sizes smaller than -2 or larger than +4, but they are treated as -2 and +4, respectively.

Working with the Font Tag

Keep in mind that the RealText `` tag works like the HTML `` tag. How you nest tags, which attributes you include, and where you place `` tags affects the outcome. For example, compare these RealText samples (**bolding** used for emphasis only):

Start with normal text.

``Make text red.

``Make red text one size larger.

``Turn off larger size for red text.

``Turn off color.

Start with normal text.

``Make text red.

``Turn off color.

``Make text one size larger.

``Turn off larger size.

Start with normal text.

``Make text red and two sizes larger.

``Make text blue and one size smaller.

``Turn off color but keep text the same size.

``Reduce text to normal.

Command Tags

The following sections describe tags you can use to launch URLs in a browser or RealPlayer. You can also use tags to issue RealPlayer commands such as **Pause** and **Play**.

Tip

Link text is the color specified in the link attribute of the `<window>` tag. The link is underlined unless the `<window>` tag includes `underline_hyperlinks="false"`.

Additional Information

SMIL files can also define hypertext links that may override the link you set here. For more information, see the SMIL chapter in *RealSystem Production Guide*.

Creating a Mail Link

This tag turns the enclosed text into an e-mail hyperlink:

```
<a href="mailto:address">...</a>
```

When the viewer clicks the link, RealText passes the e-mail address to the viewer's browser. Use an address in the standard form, such as `name@company.com`. If the browser is configured for e-mail, the e-mail client opens a new message with the defined address in the "to" line.

Opening a URL in RealPlayer or a Browser

The following RealText tag makes the text enclosed between `<a href...>` and `` a hyperlink that opens in a browser or RealPlayer:

```
<a href="URL" [target="_player"]>...</a>
```

The specified URL should begin with a protocol designation such as `http://` or `rtsp://`. The URL can use any protocol RealPlayer or the user's Web browser supports. The optional `target="_player"` attribute launches the new stream in the current RealPlayer window. If you do not use the target attribute or you specify `target="_browser"`, the linked URL opens in a Web browser window.

Example 1: Opening a URL in RealPlayer

The following example launches a new SMIL Presentation in RealPlayer:

```
<a href="rtsp://realserver.company.com/video2.smil target="_player">Play Next</a>
```

Example 2: Opening a URL in a Web Browser

This example opens a URL in the user's browser:

```
<a href="http://realguide.real.com">Visit RealGuide</a>
```

For static files, you can also specify URLs relative to the location of the RealText source file. For example, the link `...` opens the file `more.htm` in the same directory as the RealText file. Relative links follow the standard HTML directory syntax.

Example 3: Opening a URL in the Form protocol:path

If you include `version="1.5"` (or higher if using a newer version of RealText) in the `<window>` tag, you can pass the browser a URL in the form *protocol:path* instead of *protocol://path*. Protocols using this format include those for Telnet and AOL Instant Messenger. For example, here is a RealText link that launches AOL Instant Messenger:

```
<window version="1.5"...>
...<a href="aim:goim?screenname=[name]">Send Me an Instant Message</a>...
</window>
```

Issuing RealPlayer Commands

The following tag makes the enclosed text a hyperlink that, when clicked, executes a RealPlayer command:

```
<a href="command" target="_player">...</a>
```

The commands are case-sensitive and must be enclosed in double quotes. The `target="_player"` attribute is required.

Seeking Into a Presentation

The following command instructs RealPlayer to seek to the specified time in the current text stream:

```
<a href="command:seek(time)" target="_player">
```

For example, the following instructs RealPlayer to seek to 1:35.4 in the stream:

```
<a href="command:seek(1:35.4)" target="_player">
```

Pausing a Presentation

When clicked, the following link causes RealPlayer to pause the stream:

```
<a href="command:pause()" target="_player">
```

Resuming Playback

Clicking the next link causes RealPlayer to begin or resume playing the stream:

```
<a href="command:play()" target="_player">
```

Popping Up New RealPlayer Windows

RealPlayer 7 or higher can open as many player windows as the computer CPU and memory allow. This lets you keep navigation information visible in one window, for example, while content plays in another window. A RealText hyperlink that opens a new RealPlayer window uses this format:

```
<a href="command:openwindow(name, URL, playmode=value, ...)">...</a>
```

When a viewer clicks a link with this syntax, `command:openwindow` tells RealPlayer to open a new window for the given URL, and stop the presentation in the current window. This command requires two arguments, *name* and *URL*. The *playmode* arguments are optional. You can separate arguments with a comma, but this is not required. A space may precede or follow a comma. If an argument contains characters such as commas or parentheses, enclose it in single quotation marks.

Tip

Because RealPlayer G2 does not support multiple windows, add `version="1.4"` (or higher if using a newer version of RealText) to the `<window>` tag to cause RealPlayer G2 to upgrade to the latest version of RealPlayer.

name

The mandatory *name* argument comes first. It supplies a predefined or user-defined name for the new RealPlayer window. The following table gives the values for the *name* argument.

name Argument	
Name	Function
<code>_new</code> or <code>_blank</code>	Opens a new RealPlayer window each time the viewer clicks the link. Each subsequent link named <code>_new</code> or <code>_blank</code> opens a new window as well.
<code>_self</code> or <code>_current</code>	Opens the URL in the current RealPlayer window.
<i>name</i>	Creates a new RealPlayer window with the user-defined name. A subsequent <code>openwindow</code> command using the same name opens the given URL in the same window.

URL

Following the *name* argument, the required *URL* argument gives the fully qualified URL to the clip or SMIL presentation. You must include the protocol, such as `rtsp://`, `http://`, `chttp://`, or `file://` in the URL. Relative URLs do not work.

playmode

The optional *playmode=value* argument defines the state of the new RealPlayer window that opens. A command to open a new window can have more than

one playmode argument. The following table gives the possible values for playmode.

playmode Attributes and Values		
Attribute	Value	Function
autosize	true	Puts window in autosize mode so that it minimizes to just the display window when the cursor is not over it.
	false	Opens window in compact mode so that it does not change size on a mouseover. This is the default. Note that normal view mode is available only on the main RealPlayer window, not the pop-up windows.
zoomlevel	normal	Plays clip at its normal encoded size. This is the default.
	double	Doubles clip size.
	full	Plays clip at full-screen. If the operating system does not support full-screen zoom, normal mode is used instead.
ontopwhile playing	true	Keeps window on top of other windows on the desktop.
	false	Lets user determine which windows to place on top. This is the default.

Examples of Opening New RealPlayer Windows

The following sections provide examples of how to create hyperlinks that launch new RealPlayer windows.

Targeting the Same Window with Multiple Links

The following RealText link opens a URL in a new RealPlayer window named feature. The new window is set to autosize mode:

```
<a href="command:openwindow(feature, rtsp://realserver.example.com/comedy.rm, autosize=true)">Comedy Hour</a>
```

When first clicked, this link creates a RealPlayer window named feature. If another link also targets the feature window, clicking that link starts the new URL in the feature window. Clicking the following link, for example, starts an animal program in the window running the comedy program. This link switches the window out of autosize mode, though:

```
<a href="command:openwindow(feature, rtsp://realserver.example.com/animals.rm, autosize=false)">Sharks!</a>
```

Opening Separate Windows

Each link opens a separate window if the window names are different, or you use the predefined name `_new` or `_blank`. The following links open separate autosizing windows that play on top of all other desktop windows:

```
<a href="command:openwindow(_new, rtsp://realserver.example.com/comedy.rm, autosize=true, ontopwhileplaying=true)">Comedy Hour</a>
```

```
<a href="command:openwindow(_blank, rtsp://realserver.example.com/animals.rm, autosize=true, ontopwhileplaying=true)">Sharks!</a>
```

Launching URLs in the Current Window

Use either `_current` or `_self` to open the URL in the current window:

```
<a href="command:openwindow(_current, rtsp://realserver.example.com/animals.rm)">Sharks!</a>
```

The next link plays the clip at double its encoded size:

```
<a href="command:openwindow(_self, rtsp://realserver.example.com/comedy.rm, zoomlevel=double)">Comedy Hour</a>
```

Ensuring Text Delivery under any Circumstance

Use these tags to enclose text that must be delivered to RealPlayer under any circumstance:

```
<required>...</required>
```

During extremely adverse network conditions, RealSystem will halt the presentation if necessary rather than drop the text. You can use these tags sparingly, though, because RealSystem normally ensures that very little data loss occurs in transmission.

Colors

For the RealText color options, you can use any colors available through the HTML `` tag. This includes Red/Green/Blue hexadecimal values (`#RRGGBB`), as well as these predefined color names, listed here with their corresponding hexadecimal values:

white (#FFFFFF)	silver (#C0C0C0)	gray (#808080)	black (#000000)
yellow (#FFFF00)	fuchsia (#FF00FF)	red (#FF0000)	maroon (#800000)
lime (#00FF00)	olive (#808000)	green (#008000)	purple (#800080)
aqua (#00FFFF)	teal (#008080)	blue (#0000FF)	navy (#000080)

Transparency

You can also use “transparent” as a color. For example, `` means that each following word does not have a rectangle drawn behind it. This lets you draw text on top of previous text (using the `<pos/>` tags) without “erasing” the previous text.

Note

Transparency is not currently supported as a window background color.

Coded Characters

The following table lists the character codes you can include in a RealText source file. Codes begin with an ampersand (“&”) and end with a semicolon (“;”). RealText interprets these characters the same way as popular Web browsers.

RealText Coded Character Set

Code	Displays as
<code>&lt;</code>	<code><</code>
<code>&gt;</code>	<code>></code>
<code>&amp;</code>	<code>&</code>
<code>&nbsp;</code>	(nonbreaking space)
<code>&#32;</code> to <code>&#255;</code>	Characters taken from the active character set as specified by the active <code></code> tag. The default character set is iso-8859-1, which is also known as ISO Latin 1. For a list of these characters, see the W3C reference at http://www.w3.org/MarkUp/html-spec/html-spec_13.html . See below, however, if you’re using the mac-roman character set.

Tip

The HTML version of this manual includes a JavaScript file that generates the character codes for you. Visit <http://service.real.com/help/library/encoders.html> to download the zipped HTML manual.

For example, the following RealText source text:

This is a bold tag: ``.

is displayed in a RealText window as:

This is a bold tag: "".

Using the mac-roman Character Set

Unlike HTML, RealText allows you to change character sets within a document. It then takes coded characters from the active character set.

Generally, character codes 128 and below are the same in all Western-language character sets. Those above 128 may differ, though. In the mac-roman character set, for example, ¦ is a paragraph symbol. But in iso-8859-1, this symbol is ¶.

See <http://czyborra.com/charsets/mac-roman.gif> for a GIF chart of the mac-roman upper character set. Go by this chart, rather than the W3C reference provided above if you've set and are entering coded characters of  or above. The values in the chart are in hexadecimal (base 16). The chart cell in the upper, left-hand corner equals 128 in decimal (base 10), so you can count across from there. To make a paragraph symbol when using mac-roman, for instance, you use ¦ in the RealText file because hexadecimal A6 on the chart is decimal 166.

Chapter 4

EXAMPLES

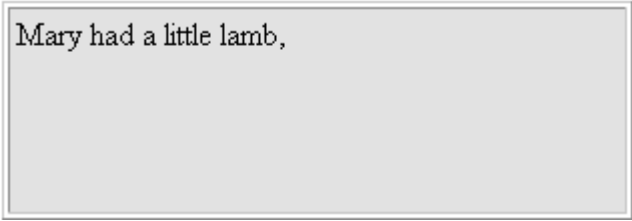
This following examples show how to create and tag the supported RealText window types.

Generic Window

The following example illustrates a RealText source document and resulting display for a generic RealText window. This is the RealText source file (extension .rt):

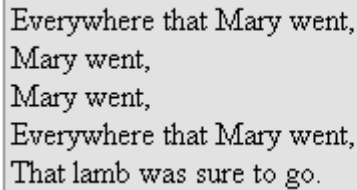
```
<window duration="30" bgcolor="yellow">
Mary had a little lamb,
<br/><time begin="3"/>little lamb,
<br/><time begin="6"/>little lamb.
<br/><time begin="9"/>Mary had a little lamb,
<br/><time begin="12"/>whose fleece was white as snow.
<br/><time begin="15"/><clear/>Everywhere that Mary went,
<br/><time begin="18"/>Mary went,
<br/><time begin="21"/>Mary went,
<br/><time begin="24"/>Everywhere that Mary went,
<br/><time begin="27"/>That lamb was sure to go.
</window>
```

When RealPlayer processes this file, it displays only the first line of the text from zero to three seconds into the stream:



Every three seconds after the first line displays, a new line appears as specified by `<time begin="seconds"/>`. At 15 seconds, `<clear/>` clears the displayed text

and resets the text “cursor” to the top, left-hand corner of the window. When the stream finishes, all lines of text following the last <clear/> tag appear in the window:



```
Everywhere that Mary went,
Mary went,
Mary went,
Everywhere that Mary went,
That lamb was sure to go.
```

Note the following about this example:

- Because it was not specified in the <window> tag, word wrapping defaults to true. However, word wrapping is not necessary because
 tags force line breaks.
- <time/> tags need not appear after a
 tag. They can appear anywhere in the text.
- The example could have used <time end="time"/> tags to make individual lines of text disappear before the <clear/> tag cleared all the lines.

TickerTape Window

The following example shows the RealText source document and resulting display for a TickerTape window. This is the RealText (.rt) source file:

```
<window type="tickertape" duration="1:00" width="350" loop="true"
underline_hyperlinks="false" link="white">
<br/><b>
<tu><a href="http://www.dowjones.com/">DJIA</a></tu>
<tl>7168.35 +36.52 </tl>
<tu>NIKEI 225 Index</tu>
<tl>20603.71 +203.11</tl>
</b>
</window>
```

This source file produces the following window in RealPlayer.



DJIA	NIKEI 225 Index
7168.35 +36.52	20603.71 +203.11

Note the following about this example:

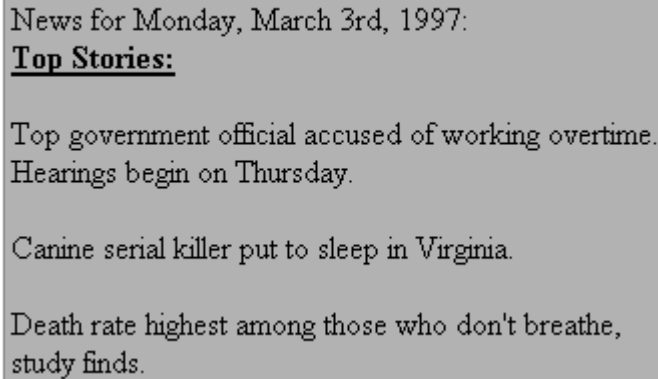
- The text crawls from right to left at 20 pixels per second, the default crawlrate for a TickerTape window.
- The `` tag at the start **bolds** all following text.
- `DJIA` makes DJIA a hyperlink that, when clicked, opens a browser to **http://www.dowjones.com/**.
- DJIA is not underlined because `underline_hyperlinks="false"` is declared in the `<window>` tag. It is drawn in white because `link="white"` is also in the `<window>` tag.
- The attribute `loop="true"` in the `<window>` tag means the text loops around and comes back in from the right side of the window as soon as the last character of the text has moved completely out of the window. It is not necessary to specify this attribute explicitly, because in TickerTape windows `loop="true"` is the default.
- The `
` tag that comes before the first text item forces the text that follows to start just past the window's right edge. Any break or paragraph tag inside TickerTape text causes the text that follows to start at the right edge. If the `
` tag were absent, the data would appear starting at the window's left edge.

Scrolling News Window

The following example demonstrates a ScrollingNews window. This is the RealText (.rt) source file:

```
<window type="scrollingnews" duration="10" bgcolor="aqua">
<br/><br/>News for Monday, March 3rd, 1997:
<br/><b><u>Top Stories:</u></b>
<p>Top government official accused of working overtime.
Hearings begin on Thursday.</p>
<time begin="1"/>Canine serial killer put to sleep in Virginia.
<p>Death rate highest among those who don't breathe, study finds.</p>
</window>
```

This source file produces the following window in RealPlayer.



News for Monday, March 3rd, 1997:
Top Stories:
 Top government official accused of working overtime.
 Hearings begin on Thursday.
 Canine serial killer put to sleep in Virginia.
 Death rate highest among those who don't breathe,
 study finds.

The example above shows what the text box looks like 1.5 seconds into the presentation. The text window and text appear at the 0,0 (top left) coordinates on the screen. At one second into the stream, the second and third items appear as specified in the `<time begin="1"/>` tag.

Note that the second item becomes visible after it scrolls into the window. This demonstrates how you can control the visibility of text with the `<time/>` tag. Without the `<time/>` tag, the text would have appeared before one second and would have scrolled up from the bottom of the window.

TelePrompter Window

The following example demonstrates a TelePrompter window. This is the RealText (.rt) source file:

```
<window type="teleprompter" height="64" duration="25"
bgcolor="lime" wordwrap="false">
<font face="system">
Out, out, brief candle!
<br/><time begin="3.5"/>Life's but a walking shadow, a poor player
<br/><time begin="7"/>That struts
<time begin="8"/>and frets
<time begin="9"/>his hour upon the stage
<br/><time begin="12"/>And then is heard no more:
<time begin="15"/>it is a tale
<br/><time begin="16"/>Told by an idiot,
```

```
<time begin="17.5"/>full of sound and fury,  
<br/><time begin="20"/>Signifying  
<time begin="22"/><font color="red">nothing.</font></font>  
</window>
```

When the window fills with text and a new line appears, all lines scroll up to make room for the new line. The following illustrates the window when the presentation ends.

That struts and frets his hour upon the stage
And then is heard no more: it is a tale
Told by an idiot, full of sound and fury,
Signifying nothing.

Note the following about TelePrompter windows:

- The wordwrap attribute can be true or false.
- The scrollrate and crawlrate attributes are ignored.
- You can use a <clear/> tag to clear the window and start the next line at the window's upper, left-hand corner.
- Use <time begin/> tags at the start of each line and do not let word wrapping cause too many line breaks between <time/> tags.
- Multiple lines of text with the same begin time cause the preceding text to move up until all new lines appear at the bottom of the window.

SUMMARY OF WINDOW TAG ATTRIBUTES



Attribute	Function	Default	Notes
bgcolor="color"	Sets the window's background color.	black for TickerTape, white for others	Refer to "Colors" on page 34.
crawlrate="pixels per second"	Sets the pixels per second that the text moves horizontally.	20 for TickerTape and Marquee, 0 for others	None.
duration="dd:hh:mm:ss.xyz"	Specifies time that presentation stops.	60 seconds	Only seconds ("ss") are required.
extraspaces="use ignore"	Recognizes or ignores extra spaces in text.	use	ignore value not used with <pre>...</pre>
height="pixels"	Sets the window height in pixels.	30 for TickerTape and Marquee, 180 for others	SMIL can override window height setting.
link="color"	Sets the color of hyperlinks.	blue	Refer to "Colors" on page 34.
loop="true false"	Determines whether looping occurs.	true for TickerTape and Marquee, false for others	Used only for TickerTape and Marquee.
scrollrate="pixels per second"	Sets the pixels per second that the text moves vertically.	10 for ScrollingNews, 0 for others	No effect on TickerTape or Marquee.
type="window type"	Sets the window type.	generic	Other values are tickertape, marquee, scrollingnews, and teleprompter.
underline_hyperlinks="true false"	Determines whether hyperlinks are <u>underlined</u> .	true	None.
version="version number"	Sets RealText version.	1.0	Required for some character sets.

(Table Page 1 of 2)

Attribute	Function	Default	Notes
width="pixels"	Determines the window width in pixels.	500 for TickerTape and Marquee, 320 for others	SMIL can override window width setting.
wordwrap="true false"	Specifies whether word wrap is performed.	true	No effect on TickerTape or Marquee.

(Table Page 2 of 2)

Appendix B

SUMMARY OF TEXT TAGS

Tag	Attributes	Function	Notes
<code>...</code>	(none)	Creates hyperlink that executes a command.	Target required.
<code>...</code>	name, URL, playmode (see page 32)	Opens new, named windows for URL	Accepts several playmode attributes.
<code>...</code>	(none)	Opens e-mail message.	Browser must be configured for e-mail.
<code>...</code>	target= " _player"	Creates hyperlink to specified URL.	URL opens in browser if target not set to player.
<code>...</code>	(none)	Bolds the enclosed text.	Compatible with HTML.
<code>
</code>	(none)	Adds space between text.	In TickerTape and Marquee, moves "cursor" to the window's right edge.
<code><center>...</center></code>	(none)	Centers the enclosed text.	Compatible with HTML.
<code><clear/></code>	(none)	Clears all text from the window.	Following text is displayed at the window's normal starting point.

(Table Page 1 of 3)

Tag	Attributes	Function	Notes
...	bcolor	Sets the background color.	See "Colors" on page 34.
	charset	Specifies character set used to display text.	Default is "us-ascii".
	color	Controls font color except for TickerTape window.	Supports all color values available in HTML. See also <tl> and <tu>.
	face	Sets the text face.	Default is "Times New Roman".
	size	Sets relative font size.	Values are: -2, -1, +0, +1, +2, +3, +4; or 1, 2, 3, 4, 5, 6, 7.
<hr/>	(none)	Acts like two tags.	For HTML compatibility.
<i>...</i>	(none)	<i>Italicizes</i> the enclosed text.	Compatible with HTML.
...	(none)	Acts like a tag.	For HTML compatibility. No numbers or bullets.
...	(none)	Indents text.	For HTML compatibility. No numbers, however.
<p>...</p>	(none)	Adds space between text.	In TickerTape and Marquee, moves "cursor" to the window's right edge.
<pos x="pixels"/>	(none)	Positions text horizontally.	Applies only if scrollrate and crawlrate are 0 (zero).
<pos y="pixels"/>	(none)	Positions text vertically.	Applies only if scrollrate and crawlrate are 0 (zero).
<pre>...</pre>	(none)	Preserves tabs, spaces, and breaks.	Uses Courier face.
<required>...</required>	(none)	Ensures that text is delivered.	Presentation will halt if needed until text gets through.
<s>...</s>	(none)	Strikes through the enclosed text.	Compatible with HTML.
<time begin="dd:hh:mm:ss.xyz"/>	(none)	Sets time when text appears.	Only seconds ("ss") are required.
<time end="dd:hh:mm:ss.xyz"/>	(none)	Sets time when text disappears.	Only seconds ("ss") are required.

(Table Page 2 of 3)

Tag	Attributes	Function	Notes
<tl [color="color"]>...</tl>	color	Displays text at lower edge in TickerTape.	Color affects only the following <tl> elements. See also "Colors" on page 34.
<tu [color="color"]>...</tu>	color	Displays text at upper edge in TickerTape.	Color affects only the following <tu> elements. See also "Colors" on page 34.
<u>...</u>	(none)	<u>Underlines</u> the enclosed text.	Compatible with HTML.
...	(none)	Indents text.	For HTML compatibility. No bullets, however.

(Table Page 3 of 3)